



TITLE:

同次対称錐計画問題の内点許容解を求める新しいアルゴリズム (数理解析研究所講究録 2018, 2069: 179-193) 最適化の発展: モデル化とアルゴリズム)

AUTHOR(S):

村松, 正和; ロウレンソ, ブルノ F.; 北原, 知就; 土谷, 隆

CITATION:

村松, 正和 ...[et al]. 同次対称錐計画問題の内点許容解を求める新しいアルゴリズム (数理解析研究所講究録 2018, 2069: 179-193)

ISSUE DATE:

2018-04

URL:

<http://hdl.handle.net/2433/241979>

RIGHT:

同次対称錐計画問題の内点許容解を求める新しいアルゴリズム

電気通信大学 村松正和¹
成蹊大学 プルノ・F・ロウレンソ²
東京工業大学 北原知就³
政策研究大学院大学 土谷隆⁴

概要

対称錐計画は線形計画(LP), 2次錐計画(SOCP), 半正定値計画(SDP) をすべて含む大きな錐線形計画のクラスである。特に、主双対内点法が適用できるほぼ最大のクラスとして知られている[8]。近年、Chubanov は[1]において、同次線形計画問題の内点許容解を求める新しいアルゴリズムを発表した。本稿ではこのアルゴリズムの対称錐計画への拡張[7]について解説する。まず1節で対称錐計画について説明し、2節でChubanovの方法の拡張について述べ、最後に関連する話題について述べる。

1 ユークリッド Jordan 代数と対称錐計画

1.1 Jordan 代数とユークリッド内積

実ベクトル空間 \mathbb{E} は、任意の元 $x, y \in \mathbb{E}$ に対して以下の性質をもつ乗算 \circ が定義されているとき **Jordan 代数** と呼ばれる:

$$x \circ y = y \circ x \quad (1)$$

$$x \circ (x^2 \circ y) = x^2 \circ (x \circ y). \quad (2)$$

ここで、 x^2 は $x \circ x$ を表している。この乗算に関する単位元を e とする。任意の $x \in \mathbb{E}$ に対して $x \circ e = e \circ x = x$ が成り立つ。また、 $x \circ y = e$ となる y を x^{-1} と表記する。

例 1 1. $\mathbb{E} = \mathbb{R}^n$ において

$$x \circ y = \begin{pmatrix} x_1 y_1 \\ \vdots \\ x_n y_n \end{pmatrix} \quad (3)$$

と定義すれば、これは (1), (2) を満たす。単位元は $\bar{e} = (1, \dots, 1)^T$ である。

2. $\mathbb{E} = \mathbb{R}^n$ において

$$x \circ y = \begin{pmatrix} x_0 \\ \tilde{x} \end{pmatrix} \circ \begin{pmatrix} y_0 \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} x^T y \\ x_0 \tilde{y} + y_0 \tilde{x} \end{pmatrix}. \quad (4)$$

と定義しても (1), (2) を満たす。ここで \tilde{x}, \tilde{y} は $n-1$ 次元ベクトルである。この乗算の単位元は $e = (1, 0, \dots, 0)^T$ である。

¹MasakazuMuramatsu@uec.ac.jp

²lourengo@st.seikei.ac.jp

³kitahara.t.ab@m.titech.ac.jp

⁴tsuchiya@grips.ac.jp

3. $n \times n$ 対称行列の集合 S^n の場合を考える。 S^n は $n(n+1)/2$ 次元の実ベクトル空間である。 $X, Y \in S^n$ 通常の積 $(XY)_{ij} = \sum_{k=1}^n X_{ik}Y_{kj}$ は (1) を満たさない。しかし

$$X \circ Y = \frac{XY + YX}{2} \quad (5)$$

と定義すれば (1), (2) を満たす。単位元は恒等行列 I である。

Jordan 代数 \mathbb{E} は、そこに以下の性質を持つ内積 $\langle \cdot, \cdot \rangle$ が定義されているとき、**ユークリッド Jordan 代数**と呼ばれる:

$$\forall x, y, z \in \mathbb{E}, \langle x \circ y, z \rangle = \langle y, x \circ z \rangle. \quad (6)$$

この性質を持つ内積をユークリッド内積と呼ぶ。以下では \mathbb{E} はユークリッド Jordan 代数を表すものとする。 \mathbb{E} において、ユークリッド内積を用いたノルムを

$$\|x\|_J = \sqrt{\langle x, x \rangle}$$

で定義する。

ベクトル $a_1, \dots, a_m \in \mathbb{E}$ に対して一次写像 $A: \mathbb{E} \rightarrow \mathbb{R}^m$ を

$$(A(x))_i = \langle a_i, x \rangle$$

で定義する。便宜上、 $A(x)$ を行列のように Ax と書くことにする。また、

$$\begin{aligned} \ker A &= \{x \in \mathbb{E} : Ax = 0\} = \{x \in \mathbb{E} : \langle a_i, x \rangle = 0 \ (i = 1, \dots, m)\}, \\ \operatorname{Im} A^T &= \left\{ y \in \mathbb{E} : \exists u \in \mathbb{R}^m \text{ s.t. } y = \sum_{i=1}^m u_i a_i \right\} \end{aligned}$$

と書く。線形代数の基本定理として、 $\ker A$ と $\operatorname{Im} A^T$ がお互いに直交補空間となることは容易に確かめられる。

\mathbb{E} において $c \circ c = c$ が成り立つ元を**射影子 (idempotent)** と呼ぶ。 \mathbb{E} の任意の元 x に対し、 r 個の射影子 c_1, \dots, c_r および r 個の実数 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$ が存在して

$$x = \lambda_1 c_1 + \dots + \lambda_r c_r$$

と書けることが知られている。ただし、射影子の集合 $\{c_1, \dots, c_r\}$ は

$$c_i \circ c_j = 0 \ (i \neq j), \quad \sum_{j=1}^r c_j = e, \quad \langle c_j, c_j \rangle = 1 \ (j = 1, \dots, r)$$

を満たす。 $\{c_1, \dots, c_r\}$ は x の **Jordan 枠 (Jordan frame)** と呼ばれる。 $\lambda_1, \dots, \lambda_r$ は x の**固有値 (Eigenvalue)**, r は \mathbb{E} の**ランク (rank)** と呼ばれる。以下では、 x の固有値を $\lambda_j(x)$ ($j = 1, \dots, r$) と書くことにする。ただし、 $\lambda_1(x) \geq \lambda_2(x) \geq \dots \geq \lambda_r(x)$ が成り立っているものと仮定する。次の関係式は重要である:

$$\langle e, x \rangle = \left\langle \sum_{i=1}^r c_i, \sum_{j=1}^r \lambda_j(x) c_j \right\rangle = \sum_{j=1}^r \lambda_j(x).$$

例 2 1. $\mathbb{E} = \mathbb{R}^n$ において (3) で \circ が定義されているとき,

$$\langle x, y \rangle = \sum_{j=1}^n x_j y_j$$

と定義すれば、これは (6) を満たす。ベクトル \mathbf{x} の固有値は (順不同で) x_1, \dots, x_n であり、射影子 \bar{e}_j に対して

$$\mathbf{x} = \sum_{j=1}^n x_j \bar{e}_j$$

が成り立つ。よってこの場合ランクは n である。

2. $\mathbb{E} = \mathbb{R}^n$ において (4) で \circ が定義されているとき、通常の内積 $\sum_{j=1}^n x_j y_j$ は (6) を満たさない。内積を

$$\langle \mathbf{x}, \mathbf{y} \rangle = 2 \sum_{j=1}^n x_j y_j$$

と定義すれば (6) は満たされる。よって $\|\mathbf{x}\|_J = \sqrt{2}\|\mathbf{x}\|$ であり、特に $\|e\|_J = \sqrt{2}$ である。ベクトル $\mathbf{x} = (x_0, \bar{\mathbf{x}}) \in \mathbb{R}^n$ の固有値分解は

$$\mathbf{x} = \lambda_1 \mathbf{c}_1 + \lambda_2 \mathbf{c}_2$$

となる。ただし

$$\lambda_1 = x_0 + \|\bar{\mathbf{x}}\|, \lambda_2 = x_0 - \|\bar{\mathbf{x}}\|, \mathbf{c}_1 = \begin{pmatrix} 1/2 \\ \frac{\bar{\mathbf{x}}}{2\|\bar{\mathbf{x}}\|} \end{pmatrix}, \mathbf{c}_2 = \begin{pmatrix} 1/2 \\ -\frac{\bar{\mathbf{x}}}{2\|\bar{\mathbf{x}}\|} \end{pmatrix},$$

である。この場合 \mathbb{E} のランクは 2 である。

3. $X, Y \in \mathbb{S}^n$ に対して

$$\langle X, Y \rangle = \text{tr}(XY)$$

で定義された内積は (6) を満たす。よって $\|I\|_J = \sqrt{n}$ である。また、実対称行列 X には n 個の固有値 $\lambda_1, \dots, \lambda_n$ と対応する単位固有ベクトル $\mathbf{q}_1, \dots, \mathbf{q}_n$ があり、

$$X = \sum_{j=1}^n \lambda_j \mathbf{q}_j \mathbf{q}_j^T$$

と固有値分解できることはよく知られた事実である。この場合 \mathbb{S}^n のランクは n であり、 $\mathbf{q}_j \mathbf{q}_j^T$ ($j = 1, \dots, n$) が射影子となる。

1.2 対称錐

ユークリッド Jordan 代数 \mathbb{E} において、「2 乗の錐」すなわち

$$\mathcal{K} = \{ \mathbf{x}^2 : \mathbf{x} \in \mathbb{E} \}$$

を対称錐と呼ぶ。これは、固有値が全て非負である元の集合と一致することが知られている。また、対称錐の内部は、固有値が全て正である元の集合と一致する。

例 3 1. (3) の場合、2 乗の錐は

$$\{ \mathbf{x}^2 : \mathbf{x} \in \mathbb{R}^n \} = \{ \mathbf{z} : z_j = x_j^2 (j = 1, \dots, n) \} = \{ \mathbf{x} : \mathbf{x} \geq \mathbf{0} \}.$$

つまり非負象限 \mathbb{R}_+^n である。 \mathbb{R}_+^n の内部は $\text{Int } \mathbb{R}_+^n = \{ \mathbf{x} : \mathbf{x} > \mathbf{0} \} = \mathbb{R}_{++}^n$ である。

2. (4) の場合、2 乗の錐は

$$\{x^2 : x \in \mathbb{R}^n\} = \left\{ \begin{pmatrix} \|x\|^2 \\ 2x_0 \tilde{x} \end{pmatrix} : x \in \mathbb{R}^n \right\} = \left\{ \begin{pmatrix} x_0 \\ \tilde{x} \end{pmatrix} : x_0 \geq \|\tilde{x}\| \right\}.$$

この錐は 2 次錐と呼ばれる。以下これを \mathbb{L}_n で表す。内部は

$$\text{Int } \mathbb{L}_n = \left\{ \begin{pmatrix} x_0 \\ \tilde{x} \end{pmatrix} : x_0 > \|\tilde{x}\| \right\}.$$

である。

3. (5) の場合、2 乗の錐は半正定値行列の集合となる。 $X \succeq O$ で X が半正定値であることを表すとすれば、

$$\{X : X \succeq O\}$$

となる。この錐は半正定値錐と呼ばれる。以下では半正定値錐を \mathbb{S}_+^n で表す。また、その内部は正定値行列錐 \mathbb{S}_{++}^n である。

ユークリッド Jordan 代数 \mathbb{E} において、ベクトル $w \in \text{int } \mathcal{K}$ に対して $\mathbf{L}(w) : \mathbb{E} \rightarrow \mathbb{E}$, $\mathbf{Q}(w) : \mathbb{E} \rightarrow \mathbb{E}$ を

$$\mathbf{L}(w)x = w \circ x, \quad \mathbf{Q}(w) = 2\mathbf{L}(w)^2 - \mathbf{L}(w^2)$$

で定義する。 $\mathbf{Q}(w)$ は 2 次表現 (Quadratic representation) と呼ばれ、次の性質を持つことが知られている。

1. \mathbf{Q} は一次変換である。
2. $\mathbf{Q}\mathcal{K} = \mathcal{K}$, すなわち \mathbf{Q} は錐を動かさない。

1.3 対称錐の直積

対称錐 \mathcal{K} は次の性質をもつ。

1. (自己双対性) $\mathcal{K}^* = \mathcal{K}$.
2. (同次性) $\text{int}(\mathcal{K})$ の任意の 2 点 x, y に対し、 $PK = \mathcal{K}$ かつ $Px = y$ であるような \mathbb{E} の一次変換 P が存在する。

実はこの 2 つの性質をもつ錐は対称錐となることが知られている。つまり、同次自己双対錐と対称錐は同じものを指している。

対称錐はそれがいくつかの対称錐の直積で表現できないとき、**単純** と呼ばれる。単純な対称錐のいくつかを表 1 に掲げる。これ以外にも単純な対称錐は存在するが、実用的な問題に出て来るものはこの 3 つでほとんどを占めている。詳しくは [4] などを参照されたい。

以下、本稿では対称錐 \mathcal{K} が p 個の単純な対称錐の直積で書かれていることを仮定する。つまり、

$$\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_p \quad (7)$$

である。線形計画であれば全ての \mathcal{K}_i が \mathbb{R}_+ であり、2 次錐計画であれば \mathcal{K}_i は \mathbb{R}_+ のほか 2 次錐 \mathbb{L}_{n_i} をとることができる。半正定値計画では \mathcal{K}_i は \mathbb{R}_+ のほか $\mathbb{S}_+^{n_i}$ が取られる。対称錐計画は、これらの錐制約を自由に用いることができる最適化問題といえる。

表 1: 単純な対称錐の例

単純な対称錐	記号	次元	ランク	単位元
非負実数の集合	\mathbb{R}_+	1	1	1
2 次錐	\mathbb{L}_n	n	2	$\begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}$
半正定値錐	\mathbb{S}_+^n	$n(n+1)/2$	n	I

錐の分解 (7) に対応してユークリッド Jordan 代数 \mathbb{E} は

$$\mathbb{E} = \mathbb{E}_1 \times \mathbb{E}_2 \times \cdots \times \mathbb{E}_p \quad (8)$$

と分解できる。 \mathbb{E} における乗算は各部分空間 \mathbb{E}_i における乗算の直積で定義される。

以後、 \mathbb{E}_i の次元は n_i 、ランクは r_i と仮定する。 \mathbb{E} の次元は $N = \sum_{i=1}^p n_i$ 、ランクは $r = \sum_{i=1}^p r_i$ となる。各 \mathbb{E}_i の単位元を e_i とすると \mathbb{E} の単位元は

$$e = e_1 \times \cdots \times e_p$$

である。

1.4 1- ∞ ノルムと ∞ -1 ノルム

これから扱う対称錐計画問題では、全空間 \mathbb{E} が (8) のように部分空間の直積として書かれていることを仮定する。また $x \in \mathbb{E}$ に対し、 \mathbb{E}_i に対応する部分を x_i と表記する。各部分ベクトル x_i はベクトルのときもあれば対称行列のときもある。

この構造を反映して、以下のノルムを定義する：

$$\|x\|_{1,\infty} = \max_{i=1,\dots,p} \|x_i\|_1,$$

ただし、

$$\|x_i\|_1 = \sum_{j=1}^{r_i} |\lambda_j(x_i)|$$

は固有値に関する 1 ノルムである。また、 $\|\cdot\|_{1,\infty}$ と双対なノルムは

$$\|x\|_{\infty,1} = \sum_{i=1}^p \max \{ |\lambda_j(x_i)| : j = 1, \dots, r_i \}$$

と定義される。

これらのノルムに関して以下の性質があることはすぐに確かめられる。

補題 1 1. (Cauchy-Schwarz の不等式)

$$\forall x, y \in \mathbb{E}, \quad |\langle x, y \rangle| \leq \|x\|_{1,\infty} \|y\|_{\infty,1}.$$

2. $x \in \mathcal{K}$ のとき、 $\|x\|_{1,\infty} = \max \{ \langle e_i, x_i \rangle : i = 1, \dots, p \}$.

3. $\|x\|_{\infty,1} \leq \sqrt{p} \|x\|_J$

例 4 1. 全ての $\mathcal{K}_1, \dots, \mathcal{K}_p$ が \mathbb{R}_+ の場合,

$$\|\mathbf{x}\|_{1,\infty} = \max \{ |x_i| : i = 1, \dots, p \} = \|\mathbf{x}\|_\infty$$

である。

2. 全ての $\mathcal{K}_1, \dots, \mathcal{K}_p$ が 2 次錐の場合,

$$\|\mathbf{x}\|_{1,\infty} = \max \{ |\lambda_1^i| + |\lambda_2^i| : i = 1, \dots, p \}$$

である。特に $\mathbf{x} \in \mathcal{K}$ の場合には

$$\|\mathbf{x}\|_{1,\infty} = \max \{ 2(\mathbf{x}_i)_0 : i = 1, \dots, p \}$$

となる。ここで $(\mathbf{x}_i)_0$ は部分ベクトル \mathbf{x}_i の先頭要素を表す。

2 同次対称錐計画問題の内点許容解を求めるアルゴリズム

2.1 扱う問題

$\mathcal{K} = \mathcal{K}_1 \times \dots \times \mathcal{K}_p$ を単純な対称錐 p 個の直積として、次の 2 つの問題を考える:

$$\langle P(A) \rangle \text{ find } \mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{0}, \mathbf{x} \in \text{int } \mathcal{K}$$

$$\langle D(A) \rangle \text{ find } (\mathbf{u}, \mathbf{y}) \text{ s.t. } \mathbf{y} = A^T \mathbf{u}, \mathbf{y} \in \mathcal{K}, \mathbf{y} \neq \mathbf{0}$$

線形計画 $\langle LP(A) \rangle$, $\langle DLP(A) \rangle$ の場合と同様に、 $\langle P(A) \rangle$ と $\langle D(A) \rangle$ はどちらか一方のみに必ず解が存在する、いわゆる「二者択一」の関係にある。

補題 2 次のうちどちらか一方のみが常に成り立つ:

1. $\langle P(A) \rangle$ に許容解がある。

2. $\langle D(A) \rangle$ に許容解がある。

我々の目的は、どちらに解が存在するかを見出すことにある。

$\langle P(A) \rangle$ に関連して、以下の問題を考える:

$$\text{find } \mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{0}, \|\mathbf{x}\|_{1,\infty} \leq 1, \mathbf{x} \in \text{int } \mathcal{K}.$$

1-inf ノルムの性質から、この問題は以下の問題と等価である:

$$\langle P_s(A) \rangle \text{ find } \mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{0}, \langle \mathbf{e}_i, \mathbf{x}_i \rangle \leq 1 \ (i = 1, \dots, p), \mathbf{x} \in \text{int } \mathcal{K}.$$

明らかに、 $\langle P_s(A) \rangle$ の許容解は $\langle P(A) \rangle$ の許容解であるし、 $\langle P(A) \rangle$ に許容解 \mathbf{x} があれば、 $\mathbf{x}/\langle \mathbf{e}, \mathbf{x} \rangle$ は $P_s(A)$ の許容解である。つまり、 $\langle P(A) \rangle$ の許容解の有無は $\langle P_s(A) \rangle$ の許容解の有無と一致する。以下では $\langle P_s(A) \rangle$ の許容集合を $\mathcal{F}_s(A)$ と置く。

$\mathbf{a}, \mathbf{b} \in \mathbb{E}_i$ に対し、半空間 $H(\mathbf{a}, \mathbf{b}) \subseteq \mathbb{E}_i$ を

$$H(\mathbf{a}, \mathbf{b}) = \{ \mathbf{x}_i \in \mathbb{E}_i : \langle \mathbf{a}, \mathbf{x}_i \rangle \leq \langle \mathbf{a}, \mathbf{b} \rangle \}$$

で定義する。一般に $\langle \mathbf{e}_i, \mathbf{e}_i \rangle = r_i$ であるので、

$$\mathbf{x} \in \mathcal{F}_s(A) \Rightarrow \mathbf{x}_i \in H(\mathbf{e}_i, \mathbf{e}_i/r_i)$$

であることに注意する。

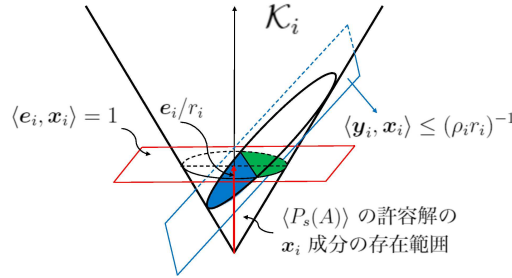


図 1: カット生成ベクトルの意味

2.2 カット生成ベクトル

次の補題は証明は簡単だが、Chubanov のアルゴリズムの対称錐への拡張において根幹をなすものである。ここで $r_{\max} = \max r_i$ である。

補題 3 ([7, Theorem 12]) ベクトル \mathbf{y} が

$$\mathbf{y} \in \mathcal{K}, \mathbf{z} = P_A \mathbf{y}, \mathbf{z} \notin \mathcal{K}, \mathbf{y} - \mathbf{z} \notin \mathcal{K}, \|\mathbf{z}\|_J \leq \frac{1}{2r_{\max}\sqrt{p}} \|\mathbf{y}\|_{1,\infty} \quad (9)$$

を満たすとする。 \mathbf{y} の i 番目の部分ベクトル $\mathbf{y}_i \in \mathbb{R}^{n_i}$ に対し、

$$\rho_i = \frac{\langle \mathbf{e}_i, \mathbf{y}_i \rangle}{r_i \sqrt{p} \|\mathbf{z}\|_J} \quad (10)$$

と定義する。もし、ある i について $\rho_i \geq 2$ ならば、任意の $\mathbf{x} \in \mathcal{F}_s(A)$ に対して $\mathbf{x}_i \in H(\mathbf{y}_i, (\rho_i r_i)^{-1} \mathbf{e}_i)$ が成り立つ。

証明: 条件より、 $P_A \mathbf{y} \neq \mathbf{0}$ である。 \mathbf{x} が $\langle P_s(A) \rangle$ の解なので、 $\langle \mathbf{e}_i, \mathbf{x}_i \rangle \leq 1$ ($i \in \{1, \dots, p\}$)、すなわち $\|\mathbf{x}\|_{1,\infty} \leq 1$ が成り立つ。補題 1 を用いると

$$\begin{aligned} \langle \mathbf{y}_i, \mathbf{x}_i \rangle &\leq \langle \mathbf{y}, \mathbf{x} \rangle = \langle \mathbf{y}, P_A \mathbf{x} \rangle = \langle P_A \mathbf{y}, \mathbf{x} \rangle \leq \|P_A \mathbf{y}\|_{\infty,1} \|\mathbf{x}\|_{1,\infty} \\ &\leq \|P_A \mathbf{y}\|_J \sqrt{p} = (r_i \rho_i)^{-1} \langle \mathbf{e}_i, \mathbf{y}_i \rangle \end{aligned}$$

が得られ、従って $\mathbf{x}_i \in H(\mathbf{y}_i, (r_i \rho_i)^{-1} \mathbf{e}_i)$ が得られる。 \square

図 1 に補題 3 の意味するところを示す。図 1 において、もともとの $\langle P_s(A) \rangle$ の許容解の \mathbf{x}_i 成分の存在範囲が錐 \mathcal{K}_i と赤い平面より下側の領域であるのに対し、補題は本当の \mathbf{x}_i の存在範囲が、青い面で切られた小さな部分となっていることを表している。このようにして、存在範囲の体積を狭めることができる。

(9) を満たすベクトル \mathbf{y} を **カット生成ベクトル** と呼ぶ。

2.3 Main Algorithm の考え方

さて、 $\langle LP(A) \rangle$ の場合、Basic Procedure からカット生成ベクトルが得られれば、対応する A の列を 2 倍することによってスケールリングを変更した。対称錐計画の場合、これに対応する操作には 2 次表現を用いる。

今、Main Algorithm の k 番目の反復で Basic Procedure がカット生成ベクトル y を返して来たとする。この y および $J = \{i : \rho_i \geq 2\}$ に対し、以下のように定める：

$$\begin{aligned}\alpha_i &= \left(\frac{1}{\rho_i} - \frac{1}{\sqrt{\rho_i(3\rho_i - 2)}} \right), \\ \beta_i &= r_i - \alpha_i, \\ w_i &= \frac{r_i \rho_i \alpha_i}{\langle e_i, y_i \rangle} y_i + \beta_i e_i, \\ v_i &= w_i^{-1}.\end{aligned}$$

すると、次が成立する。

定理 4 $i \in J$ に対し、上記のように w_i, v_i を定めると、次が成り立つ。

1. $H(y_i, (\rho_i r_i)^{-1} e_i) \cap H(e_i, r_i^{-1} e_i) \subseteq H(w_i, v_i)$.
2. $r_i Q(w_i^{-1/2})(H(e_i, e_i/r_i)) = H(w_i, v_i)$.

ここで $w_i^{-1/2}$ は 2 乗すると w_i^{-1} となるベクトル、 $Q(w_i^{-1/2})$ は $w_i^{-1/2}$ の 2 次表現である。 $w_i^{-1/2} \in \mathcal{K}_i$ より、 $Q(w_i^{-1/2})$ は正則であり、また $Q(w_i^{-1/2})\mathcal{K}_i = \mathcal{K}_i$ である。

この定理の意味するところを図 2 に示す。図 2 の左図において、青と緑の面と錐の境界で囲まれた部分は、カット生成ベクトル y により制限された x_i の存在範囲である。 $H(w_i, v_i) = \{x_i : \langle w_i, x_i \rangle \leq r_i^{-1}\}$ はこの集合を含むような半空間となっている。そして、一次変換 $r_i^{-1} Q(w_i^{1/2})$ は

1. \mathcal{K} を \mathcal{K} に写し、
2. $\langle w_i, x_i \rangle \leq r_i^{-1}$ を $\langle e_i, x_i \rangle \leq 1$ に写す (図 2 の右)。

この一次変換を用いて、Main Algorithm の反復を以下のように考える。 k 番目の反復で Basic Procedure がカット生成ベクトル y を返したとする。 $\langle P_s(A^k) \rangle$ の任意の $x \in \mathcal{F}_s(A^k)$ について、 $i \in J$ に対して $x_i \in H(w_i, v_i)$ が成り立つ。したがって $\langle P_s(A^k) \rangle$ は次の問題と同値である：

$$\langle P'_s \rangle A^k x = 0, x \in \mathcal{K}, \sum_{i=1}^p \langle e_i, x_i \rangle \leq 1, x_i \in H(w_i, v_i) \ (i \in J).$$

ブロック対角行列 $Q = \text{Diag}(Q_1, \dots, Q_p)$ を以下のように定める：

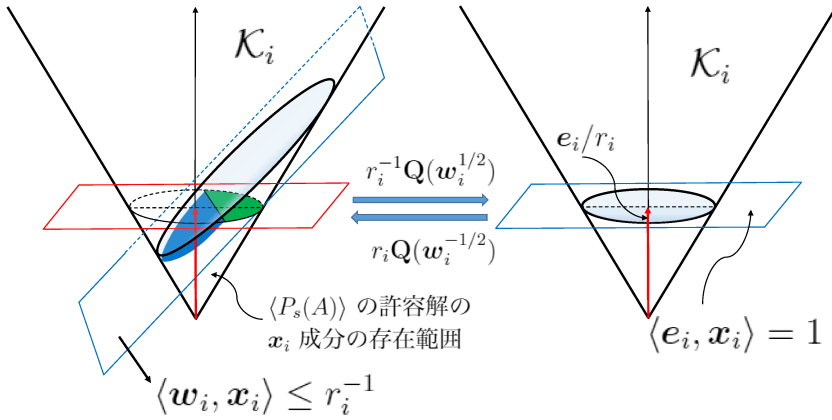
$$Q_i = \begin{cases} I & \text{if } i \notin J \\ r_i Q(w_i^{-1/2}) & \text{if } i \in J. \end{cases}$$

ここで $x = Qx'$ と変数変換すると、 $\langle P'_s \rangle$ は

$$\langle P''_s \rangle A^k Qx' = 0, Qx' \in \mathcal{K}, \langle e_i, Q_i x'_i \rangle \leq 1 \ (i \in \{1, \dots, p\}), Q_j x'_j \in H(w_j, v_j) \ (j \in J)$$

と等価である。定理 4 の 2 より、 $j \in J$ のとき

$$Q_j x'_j \in H(w_j, v_j) \Leftrightarrow x' \in H(e_j, e_j/2) \Leftrightarrow \langle e_j, x' \rangle \leq 1$$

図 2: 一次変換 Q の効果

が成り立つ。 $i \notin J$ に関しては $\langle e_i, Q_i x'_i \rangle = \langle e_i, x'_i \rangle \leq 1$ がそのまま成り立つ。従って、

$$\langle e_i, x'_i \rangle \leq 1 \quad (i = 1, \dots, p)$$

となる。さらに $i \in J$ に関しては $\langle e_i, Q_i x'_i \rangle = \langle Q_i e_i, x'_i \rangle = 2 \langle w_i, x'_i \rangle$ より

$$\langle e_i, Q_i x'_i \rangle \leq 1 \Leftrightarrow \langle w_i, x'_i \rangle \leq 1/r_i$$

であるので、まとめると $\langle P''_s \rangle$ は

$$\langle P''_s \rangle \quad A^k Q x' = 0, x' \in K, \langle e_i, x'_i \rangle \leq 1 \quad (i = 1, \dots, p), \langle w_i, x'_i \rangle \leq 1/r_i \quad (i \in J)$$

と等価である。

$A^{k+1} = A^k Q$ と置けば

$$\langle P_s(A^{k+1}) \rangle \quad A^{k+1} x' = 0, x' \in K, \langle e_i, x'_i \rangle \leq 1 \quad (i = 1, \dots, p)$$

は $\langle P'''_s \rangle$ の緩和問題となっている。つまり、 $\langle P_s(A^k) \rangle$ に解があるならば、 $\langle P_s(A^{k+1}) \rangle$ にも解がある。数学的帰納法を用いると、 $\langle P_s(A^1) \rangle = \langle P_s(A) \rangle$ に解があるならば、 $\langle P_s(A^k) \rangle$ に解があることが証明できる。

Main Algorithm の全体を図 3 に記す。また、表 2 は、Main Algorithm が A と $\epsilon > 0$ が与えられたときに返す値を記したものである。

2.4 体積に関する考察と Main Algorithm の反復回数

Main Algorithm の収束の証明において鍵となるのは、 $H(w_i, v_i) \cap K_i$ の体積が $H(e_i, e_i/r_i) \cap K_i$ の体積と比べて（一定量）小さくなる、という事実である。体積が小さいと、それに応じて最小固有値も小さくなければならない。この性質を利用して、反復が十分に進んだ場合に

$\langle P_s(A) \rangle$ の任意の許容解は ϵ よりも小さな固有値を持つ

表 2: Main Algorithm の返り値とその意味

返り値	意味
$\mathbf{A}(\mathbf{y})$	\mathbf{y} は $\langle P(A) \rangle$ の許容解である。
$\mathbf{B}(\mathbf{y})$	\mathbf{y} は $\langle D(A) \rangle$ の許容解である。
D	$\langle P_s(A) \rangle$ の任意の解は最小固有値が ϵ 以下である。

```

1: procedure MAINALGORITHM( $\mathbf{a}_1, \dots, \mathbf{a}_m, \mathcal{K}, \epsilon$ )
2:    $\epsilon = 0$ 
3:   for  $k = 1 : \text{maxiter}$  do
4:     Call BasicProcedure with  $(\mathbf{a}_1, \dots, \mathbf{a}_m)$ 
5:     if  $\mathbf{A}(\mathbf{y})$  is returned then
6:       while  $k > 0$  do
7:          $\mathbf{y} \leftarrow Q^k \mathbf{y}$ 
8:          $k \leftarrow k - 1$ 
9:       end while
10:      return  $\mathbf{A}(\mathbf{y})$ 
11:     else if  $\mathbf{B}(\mathbf{y})$  is returned then
12:       while  $k > 0$  do
13:          $\mathbf{y} \leftarrow (Q^k)^{-1} \mathbf{y}$ 
14:          $k \leftarrow k - 1$ 
15:       end while
16:      return  $\mathbf{B}(\mathbf{y})$ 
17:     end if
18:     % これより下では  $\mathbf{y}$  はカット生成ベクトル.
19:     for  $i \in \{1, \dots, p\}$  do
20:       Compute  $\rho_i$  by (10)
21:       if  $\rho_i \geq 2$  then
22:          $\alpha \leftarrow \frac{1}{\rho_i} - \frac{1}{\sqrt{\rho_i(3\rho_i - 2)}}$ 
23:          $\beta \leftarrow r_i - \alpha$ 
24:          $\mathbf{w}_i \leftarrow \frac{\alpha r_i \rho_i}{\langle \mathbf{e}, \mathbf{y} \rangle} \mathbf{y}_i + \beta \mathbf{e}_i$ 
25:          $\epsilon[i] \leftarrow \epsilon[i] + \log r_i - r_i^{-1} \log \det(\mathbf{w}_i)$ 
26:         if  $\epsilon[i] < \log r_i + \log \epsilon$  then
27:           return C
28:         end if
29:          $Q_i \leftarrow r_i \mathbf{Q}(\mathbf{w}_i^{-1/2})$ 
30:       else
31:          $Q_i \leftarrow I$ 
32:       end if
33:     end for
34:      $Q^k = \text{Diag}(Q_1, \dots, Q_p)$ 
35:     for  $i = 1 : m$  do
36:        $\mathbf{a}_i \leftarrow Q^k \mathbf{a}_i$ 
37:     end for
38:   end for
39: end procedure

```

▷ 返り値は表 2 の通り
 ▷ 解の存在範囲のチェック用

▷ 少なくとも 1 つは $\rho_i \geq 2$ となるものがある

▷ 体積が減る割合の対数
 ▷ ϵ 解が存在しないことが判明.

▷ 体積を小さくできないときはそのまま

図 3: Main Algorithm (対称錐版)

ことを宣言し、Main Algorithm は終了する。

具体的な評価のために、記号を導入する。 $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{E}_i$ に対し、 $V_i(\mathbf{a}_i, \mathbf{b}_i) = \text{vol}(H(\mathbf{a}_i, \mathbf{b}_i) \cap \mathcal{K}_i)$ と定義する。関数 φ を

$$\varphi(\rho) = 2 - \frac{1}{\rho} - \sqrt{3 - \frac{2}{\rho}}$$

と定義すれば、 $i \in J$ に対して次が成り立つ [7, Theorem 12, (iii)]:

$$V_i(\mathbf{w}_i, \mathbf{v}_i) = \left(\frac{r_i}{(\det(\mathbf{w}_i))^{1/r_i}} \right)^{d_i} V_i(\mathbf{e}_i, \mathbf{e}_i/r_i) \leq \exp \left(-\frac{\varphi(\rho_i)d_i}{r_i} \right) V_i(\mathbf{e}_i, \mathbf{e}_i/r_i). \quad (11)$$

φ は $\rho \geq 1$ で単調増加関数であり、 $\rho = 1$ ならば $\varphi(\rho) = 0$ 、 $\rho = 2$ ならば $\varphi(\rho) = 2/3 - \sqrt{2} > 0$ である。また $\phi(\rho) = \exp(-\varphi(\rho))$ とすると、 $\phi(\rho)$ は $\rho \geq 1$ で単調減少関数であり、 $\phi(1) = 1$ 、 $\phi(2) = \exp(-(3/4 - \sqrt{2}/2)) \simeq 0.9580$ である。つまり、 $i \in J$ のとき、 \mathbf{Q} によりこの体積は毎回定数倍だけ減少する。

Main Algorithm では ϵ という配列を通して状態をモニタしている。この配列は、次の性質を持つ。ここで、 $\lambda_{r_i}(\mathbf{x}_i)$ は \mathbf{x}_i の最小固有値である。

補題 5 [7, Lemma 16] $\langle P_s^1 \rangle = \langle P_s \rangle$ の任意の解 \mathbf{x} に対し、Main Algorithm の任意の反復 k において次が成り立つ:

$$\epsilon[i] \geq \log r_i + \log \lambda_{r_i}(\mathbf{x}_i).$$

この補題より、次が導かれる。

系 6 もしある i において $\epsilon[i] < \log r_i + \log \epsilon$ が成り立てば、 $\langle P_s \rangle$ の任意の解 \mathbf{x} において $\lambda_{r_i}(\mathbf{x}_i) < \epsilon$ 。

これを用いて、Main Algorithm の反復回数を評価できる。

定理 7 Main Algorithm は

$$r\varphi(2)^{-1} \log \epsilon^{-1} (\leq 12r \log \epsilon^{-1})$$

回以下の反復で終了する。

証明: Basic Procedure が毎回カット生成ベクトルを返して来たとすれば、各反復で $\rho_i \geq 2$ なる添字 i が存在し、そのとき

$$\log r_i - \frac{1}{2} \log \det \mathbf{w}_i \leq -\frac{\varphi(2)}{r_i} < 0$$

が成立する。つまり、 $\epsilon[i]$ は少なくとも $\varphi(2)/r_i > 0$ だけ減少する。Corollary 6 より、ある i が

$$r_i \varphi(2)^{-1} \log \left(\frac{1}{r_i \epsilon} \right)$$

回選択されれば ϵ 解が存在しないことがわかる。従って、せいぜい

$$\sum_{i=1}^p r_i \varphi(2)^{-1} \log \left(\frac{1}{r_i \epsilon} \right) \leq r \varphi(2)^{-1} \log \left(\frac{1}{\epsilon} \right)$$

回 Basic Procedure を呼べば Main Algorithm は終了する。 □

```

1: function BASICPROCEDURE( $A = (a_1, \dots, a_m)^*$ ,  $\mathcal{K}$ )
2:    $y^0 = \frac{e}{r}$ ,  $z^0 = P_A y^0$ .
3:   for  $k = 0, 1, 2, \dots$  do
4:     if  $z^k \in \text{Int } \mathcal{K}$  then return  $A(z^k)$ 
5:     else if  $y^k - z^k \in \text{Int } \mathcal{K}$  then return  $B(y^k - z^k)$ 
6:     else if  $2\sqrt{\rho}\|z^k\|_J \leq r_{\max}^{-1}\|y\|_{1,\infty}$  then return  $C(y^k)$ 
7:     end if
8:     Let  $c$  be an idempotent such that  $\langle e, c \rangle = 1$  and  $\langle z, c \rangle \leq 0$ .
9:      $q \leftarrow P_A c$ 
10:     $\alpha = \frac{q^T(q - z^k)}{\|z^k - q\|_J^2}$ 
11:     $y^{k+1} = \alpha y^k + (1 - \alpha)c$ ,  $z^{k+1} \leftarrow \alpha z^k + (1 - \alpha)q$ .
12:   end for
13: end function

```

▷ 返り値は表 3 の通り

▷ z^k は $\langle P(A) \rangle$ の許容解
▷ $y^k - z^k$ は $\langle D(A) \rangle$ の許容解
▷ y^k はカット生成ベクトル

図 4: Basic Procedure (対称錐計画版)

表 3: Basic Procedure の返り値

返り値 (ベクトル)	満たされている条件
$A(y)$	$Ay = 0$, $y \in \text{int } \mathcal{K}$.
$B(y)$	$y \neq 0$, $y \in \mathcal{K} \cap \text{Im } A^T$.
$C(y)$	y はカット生成ベクトル

2.5 Basic Procedure

対称錐の場合の Basic Procedure を図 4 に示す。Basic Procedure は $a_1, \dots, a_m \in \mathbb{E}$ を受け取り、これを一次写像 A とみなして射影を計算する。具体的には、 P_A は

$$\ker A = \{x \in \mathbb{E} : \langle a_i, x \rangle = 0\}$$

への射影を表す。また、返り値としては表 3 のどれかを返す。

Basic Procedure の動きは線形計画の場合と似ている。アルゴリズムを通して、 y は常に錐に含まれ、かつ $\langle e, y \rangle = 1$ を満たしている。 y を $\ker A$ に射影し、それが条件を満たさなければ、カット生成ベクトルの条件をなるべく満たすように次の点を生成する。図 5 にその動きの概略を示す。

図 5 において、 y を $\ker A$ に射影した点 z が、 $\langle P(A) \rangle$ の許容解でも $\langle D(A) \rangle$ の許容解でもなかったとする。このとき、

$$\langle c, z \rangle \leq 0$$

を満たす射影子 c を計算することができる。その方法はいくつかあるが、例えば z の固有値分解 $z = \sum_{j=1}^r \lambda_j c_j$ において、最も小さい固有値は負のはずなので、これに対応する射影子を取れば良い。SDP の場合には、これは本質的には最小固有値に対する固有ベクトルを計算することに対応する。

この c を再び $\ker A$ に射影したものを q とする。 q がやはり $\langle P(A) \rangle$ の許容解でも $\langle D(A) \rangle$ の許容解でもなかった場合、

z と q を結ぶ直線上の点で、最も原点に近い点

を次の射影された点 z' とし、対応する c - y 上の点を次の y' とするのである。このようにして生成される点列は、次の性質をもつことが証明できる:

$$\frac{1}{\|z'\|_J^2} \geq \frac{1}{\|z\|_J^2} + 1.$$

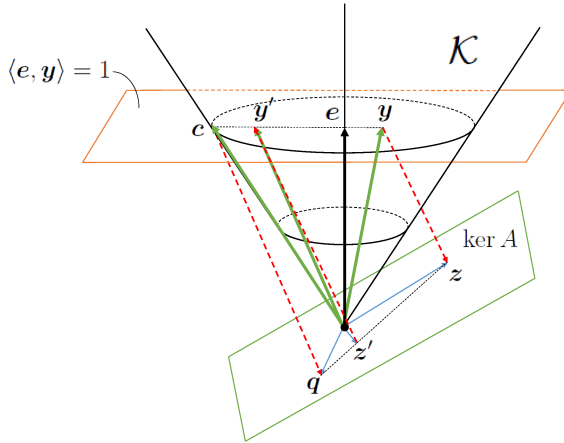


図 5: Basic Procedure の動き

この性質を利用すると、以下の定理が証明できる。

定理 8 [7, Proposition 14] *Basic Procedure* は $4p^3r_{\max}^2$ 回以下の反復で終了する。

3 関連する話題

本稿では、Chubanov の方法 [1, 2, 10] の対称錐への拡張 [7] について説明した。

論文 [7] は、2次錐計画に関する Chubanov の方法の拡張 (Kitahara and Tsuchiya [6]) に大きな影響を受けている。特に Chubanov の方法のスケーリングと2次表現をつなぎ合わせ、それらによる体積の減少を評価するというアイデアは、[6] がオリジナルである。もともと2次錐計画なので、ここから、対称錐の直積を考えることも自然に出て来た。

結果としてアルゴリズムはよりシンプルになった。つまり、[7] のアルゴリズムを2次錐計画に適用したものは [6] のアルゴリズムとは異なったものとなっている。実装した場合にどちらが良いのかはまだわかっていない。

Chubanov の方法の対称錐への拡張は、Peña ら [9] も行なっている。本稿で紹介したアルゴリズムと同様に、Main Algorithm から Basic Procedure を呼び出す手法である。その収束の解析は $\ker A \cap \mathcal{K}$ の条件数

$$\delta(\ker A \cap \mathcal{K}) = \max_x \{ \det(x) : x \in \ker A \cap \mathcal{K}, \|x\|_2^2 = r \}$$

が、アルゴリズムが進むにつれて大きくなることをベースに行われ、結果として、彼らのアルゴリズムの Main Algorithm は $\log_{1.5} \delta(\ker A \cap \mathcal{K})^{-1}$ 回の反復で終了することが示されている。また [9] では、Basic Procedure において採用できる様々な方法を提案／解析している。そのうちの一つは Soheili and Peña [11] の Smooth Perceptron Scheme である。

条件数か、錐と半空間の交わりの体積を評価するか、というそもそもの着目点の違い以外に、[9] と [7] の大きな違いは2つある。

- [9] は 1 つの対称錐をターゲットとしており、収束の係数は $r = \sum_{j=1}^p r_j$ によって表現されている。
[7] においては対称錐の直積を考えているので、 p と r_{\max} を用いて表現されている。必ずしもどちらか一方がいつも優れているわけではないが、2 次錐計画など、多くの場合では後者がより良い計算複雑度になる。
- [9] は Main Algorithm の反復が $\delta(\ker A \cap K)^{-1}$ によって表現されているが、これはあらかじめ計算することが困難な量である。それに比べ [7] では、各反復で「 $\langle P_s(A) \rangle$ の許容解の最小固有値はいくつ以下か」という情報を常に持ちつつ、アルゴリズムが動く。

2017 年になり、Chubanov は線形計画問題に対する新しいアルゴリズム [3] を発表した。これは上で述べたアルゴリズムの線形計画問題版に関して、双対側 $\langle D \rangle$ の許容解を求めるアルゴリズムであり、大きな特徴としては、

$\langle D \rangle$ の許容解かどうかを確かめるオラクルがある

ことを仮定している。与えた解が許容解でない場合、このオラクルは $A^T u$ が負になる添字を返す。射影とスケーリングを用いる点は似ているが、この研究は（オラクルを除いて） n に依存しない形で計算複雑度が決定されている。藤重 [5] はこの性質を利用して、劣モジュール関数の最小化問題に対する弱多項式アルゴリズムを導いた。

対称錐の場合に、同様なオラクルベースのアルゴリズムが構築できるかどうかは、興味深い研究テーマである。

アルゴリズムの実装は常に重要な研究課題である。特に、錐の直積を効率良く扱い、射影やスケーリングといった操作を効率良く安定的に行うためには、数値線形代数およびプログラミングに関するかなりの知識が必要である。

筆者らが Matlab で予備的に実装した範囲では、多くの場合に理論通りに動くことが確認されている。しかしながら、Main Algorithm の反復が多くなると、最終的に答えをだすときに乗算を繰り返さねばならず、ここで情報が失われることがあることも観測されている。こういうことを防ぎうるのかどうか、今後の研究を待ちたい。

謝辞

本研究は JSPS 科研費 17K00031, 15K15941, 15H02968 の助成を受けた。いくつもの美しい図を作成してくれた川端晴美氏に深く感謝する。

参考文献

- [1] S. Chubanov, “A strongly polynomial algorithm for linear systems having a binary solution”, *Mathematical Programming, Series A*, Vol. 134, 533-570 (2012).
- [2] S. Chubanov, “A polynomial projection algorithm for linear feasibility problems”, *Mathematical Programming, Series A*, Vol. 153, 687-713 (2015).
- [3] S. Chubanov, “A polynomial algorithm for linear feasibility problems given by separation oracles”, *OptimizationOnline*, 2017/01/5383.pdf (2017).

- [4] J. Faraut and A. Korányi, *Analysis on symmetric cones*, Oxford mathematical monographs, Clarendon Press, Oxford, 1994.
- [5] S. Fujishige, “A note on submodular function minimization by Chubanov’s LP algorithm”, OptimizationOnline, 2017/09/6717.pdf (2017)
- [6] T. Kitahara and T. Tsuchiya, “An extension of Chubanov’s polynomial-time linear programming algorithm to second-order cone programming”, OptimizationOnline, 2016/11/5713.pdf (2016).
- [7] B. F. Lourenço, T. Kitahara, M. Muramatsu, and T. Tsuchiya, “An extension of Chubanov’s algorithm to symmetric cones”, OptimizationOnline, 2016/12/5790.pdf (2016).
- [8] Yu. E. Nesterov and M. J. Todd, “Self-scaled barriers and interior-point methods for convex programming”, *Mathematics of Operations Research*, Vol. 22, 1-42 (1997).
- [9] J. Peña and N. Soheili, “Solving conic systems via projection and rescaling”, arXiv:1512.06154, to appear in: *Mathematical Programming, Series A*, 2017.
- [10] K. Roos, “Speeding up Chubanov’s Basic Procedure”, OptimizationOnline, 2014/09/4551.pdf (2014).
- [11] N. Soheili and J. Peña, “A smooth perceptron algorithm”, *SIAM Journal on Optimization*, Vol. 22, 728-737 (2012).
- [12] 田村明久・村松正和「最適化法」共立出版、2002年.